

Базовые типы .NET

Числовые типы

Явное преобразование типов:

(тип)выражение

При преобразовании `double` в `int` дробная часть отбрасывается: `(int)0.4`, `(int)0.9`, `(int)-0.9` все вернут значение 0.

Поля и методы любых числовых типов

```
static const числовой_тип MaxValue;
static const числовой_тип MinValue;
static числовой_тип Parse(string value[,
    IFormatProvider p]);
static bool TryParse(string value,
    out числовой_тип result);
```

В качестве параметра `p` можно использовать объект типа `System.Globalization`. Можно изменить текущие региональные настройки, используемые в программе:

```
System.Threading.Thread.CurrentThread.CurrentCulture
= new System.Globalization.CultureInfo("en-US");
```

```
string ToString([string fmt][, IFormatProvider p]);
```

Следующие классовые неизменяемые поля и методы определены только для вещественных типов. Приведем их заголовки для типа `double`.

Поля и методы вещественных числовых типов

```
static const double Epsilon;
static const double NaN;
static const double NegativeInfinity;
static const double PositiveInfinity;
static bool IsInfinity(double value);
static bool IsNaN(double value);
static bool IsNegativeInfinity(double value);
static bool IsPositiveInfinity(double value);
```

Стандартные математические функции: System.Math

`E` – константа e , 2.71828182845905

`PI` – константа π , 3.14159265358979

```
static числовой_тип Min(числовой_тип val1,
    числовой_тип val2);
static числовой_тип Max(числовой_тип val1,
    числовой_тип val2);
static числовой_тип Abs(числовой_тип value);
static int Sign(числовой_тип value);

static long BigMul(int a, int b);
static int DivRem(int a, int b, out int rem);
static long DivRem(long a, long b, out long rem);
static double Round(double value[, int d][,
    MidpointRounding mode]);
static decimal Round(decimal value[, int d][,
    MidpointRounding mode]);
```

Параметр `mode` определяет способ округления для «средних» значений: `MidpointRounding.ToEven` (по умолчанию) или `MidpointRounding.AwayFromZero`.

Для следующих методов все параметры и возвращаемые значения имеют тип `double` (для методов `Ceiling`, `Floor` и `Truncate` реализованы также варианты, в которых параметр и возвращаемое значение имеют тип `decimal`).

- `IEEERemainder(x, y)` – остаток от деления числа x на число y , равный $x - y * \text{Round}(x/y)$.
- `Ceiling(x)`, `Floor(x)`, `Truncate(x)` – округление числа x соответственно с избытком, с недостатком и по направлению к нулю.
- `Sqrt(x)` – квадратный корень из x .
- `Pow(x, y)` – степенная функция x^y .
- `Exp(x)` – показательная функция e^x .
- `Log(x)`, `Log10(x)`, `Log(x, a)` – логарифмические функции по основанию e , 10 и a соответственно: $\ln x$, $\lg x$, $\log_a x$.
- `Sin(x)`, `Cos(x)`, `Tan(x)` – тригонометрические функции; x измеряется в радианах.
- `Asin(x)`, `Acos(x)`, `Atan(x)` – обратные тригонометрические функции; возвращаемые значения измеряются в радианах и лежат в следующих промежутках: $[-\pi/2, \pi/2]$ для `Asin` и `Atan`, $[0, \pi]$ для `Acos`.
- `Atan2(y, x)` – угол наклона радиус-вектора с координатами $\{x, y\}$ к положительной полуоси OX , измеряемый в радианах и лежащий в промежутке $(-\pi, \pi]$.
- `Sinh(x)`, `Cosh(x)`, `Tanh(x)` – гиперболические функции.

Генерация случайных чисел: System.Random

```
Random([int seed]);
```

Если параметр `seed` не указан, то датчик инициализируется значением, полученным на основе текущего времени.

```
int Next([int min,] int max);
```

Значения по умолчанию: `min = 0`, `max = int.MaxValue`. Возвращает значение из полуинтервала $[min, max)$. Если `min == max`, то возвращается `min`.

```
double NextDouble();
```

Возвращает число из полуинтервала $[0, 1)$.

```
void NextBytes(byte[] buffer);
```

Заполняет все элементы массива числами из диапазона 0–255 (включая границы).

Тип char

Символы и их коды

Для определения кода по символу `c` или символа по его коду `n` надо выполнить преобразование типов: `(int)c` и `(char)n`.

Методы типа char

```
static bool IsControl(char c);
static bool IsDigit(char c);
static bool IsLetter(char c);
static bool IsLetterOrDigit(char c);
static bool IsLower(char c);
static bool IsUpper(char c);
static bool IsPunctuation(char c);
static bool IsSymbol(char c);
static bool IsSeparator(char c);
static bool IsWhiteSpace(char c);
```

```
static char ToLower(char c);
```

```
static char ToUpper(char c);
```

```
string ToString();
```

```
static string ToString(char c);
```